(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1176437 B
(45)授权公告日 2010.06.02

(21)申请号 97116048.1

(22)申请日 1997.08.15

(30)优先权数据
699294 1996.08.19 US

(73)专利权人 三星电子株式会社
地址 韩国京畿道

(72)发明人 森甬·P·桑
莫塔兹·A·穆罕默德 利·恩格延
朴宪哲

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 马莹

(51)Int.Cl.
G06F 15/163(2006.01)

审查员 韩燕

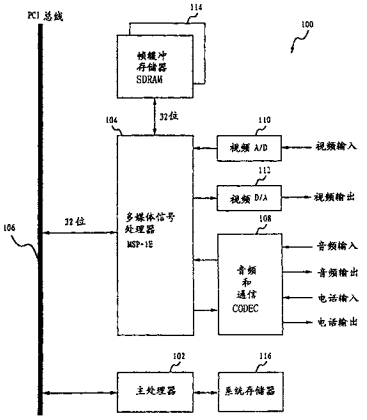
权利要求书 3 页 说明书 14 页 附图 9 页

(54)发明名称

在不对称多处理器架构中处理中断和异常的系统和方法

(57)摘要

含有多个具有控制和数据处理特性互不相同的不对称处理器的多处理器计算机系统。指令集各不相关的这些不对称处理器由单一操作系统控制。该多处理器计算机系统使用中断和异常处理的多处理器架构定义。在复位或检测到异常时，所述多处理器计算机系统的数据处理器进入空闲状态，该系统的控制处理器读和写数据处理器内的也能由数据处理器读写的各控制和状态寄存器。因此在执行操作系统或应用程序时，控制处理器控制数据处理器的运行。



CN 1176437 B

1. 一种计算机系统，包括：

用于处理不对称多处理器架构的中断和异常的多个互相连接的处理器，该多个互相连接的处理器具有互不相同的控制和数据处理特性，并具有互相独立的指令集，还包括第二处理器及第一处理器，所述第一处理器具有实际上比所述第二处理器的数据路径宽度要宽的数据路径宽度；

共同的操作系统，用于控制所述多个处理器；

中断和异常处理器，用于和所述操作系统一起运行，包括：

在所述多个处理器的第一处理器上运行的中断和异常检测器，用于检测所述第一处理器的各中断和异常条件；以及

在所述多个处理器的第二处理器上运行的中断和异常子处理器，所述中断和异常子处理器用于检测所述第一处理器的各中断和异常条件及组合地处理所述第一处理器和所述第二处理器的各中断和异常条件；

在所述多个处理器的所述第一处理器上运行的所述中断和异常检测器响应检测到的异常而将所述第一处理器去激活到空闲状态。

2. 按照权利要求 1 所述的计算机系统，其特征在于：

所述第二处理器是具有规定数据路径宽度的控制处理器；以及

所述第一处理器是向量处理器，该向量处理器具有比所述第二处理器的数据路径宽度大得多的宽数据路径宽度。

3. 按照权利要求 1 所述的计算机系统，其特征在于：

所述第二处理器进行对所述第一处理器的各控制和状态寄存器的读和写的访问。

4. 按照权利要求 3 所述的计算机系统，其特征在于：

所述第二处理器被约束在所述第一处理器处于所述空闲状态时进行对所述第一处理器的各控制和状态寄存器的读和写的访问。

5. 一种计算机系统，包括：

用于处理不对称多处理器架构的中断和异常的多个互相连接的处理器，该多个互相连接的处理器具有互不相同的控制和数据处理特性，并具有互相独立的指令集，还包括第二处理器及第一处理器，所述第一处理器具有实际上比所述第二处理器的数据路径宽度要宽的数据路径宽度；

共同的操作系统，用于控制所述多个处理器；

中断和异常处理器，用于和所述操作系统一起在所述多个处理器的第一处理器和第二处理器上运行，其中所述第二处理器有对所述第一处理器的各控制和状态寄存器的读和写访问；

在所述多个处理器的所述第一处理器上运行的所述中断和异常检测器响应检测到的异常而将所述第一处理器去激活到空闲状态。

6. 按照权利要求 5 所述的计算机系统，其特征在于所述中断和异常处理器还包括：

在所述多个处理器的所述第一处理器上运行的中断和异常检测器，用于检测所述第一处理器的各中断和异常条件。

7. 按照权利要求 5 所述的计算机系统，其特征在于所述中断和异常处理器还包括：

在所述第二处理器上运行的中断和异常子处理器，用于检测所述第一处理器的各中断

和异常条件以及用于组合地处理所述第一处理器和所述第二处理器的各中断和异常条件。

8. 按照权利要求 5 所述的计算机系统，其特征在于：

所述第二处理器是具有规定数据路径宽度的控制处理器；以及

所述第一处理器是向量处理器，该向量处理器具有比所述第二处理器的所述数据路径宽度大得多的宽数据路径宽度。

9. 按照权利要求 5 所述的计算机系统，其特征在于：

所述第二处理器被约束在所述第一处理器处于所述空闲状态时有对所述第一处理器的各控制和状态寄存器的读和写访问。

10. 一种处理不对称多处理器架构的中断和异常的方法，所述方法为一种在对含有不对称处理器的计算机系统中处理的中断和异常进行处理的方法，所述不对称处理器具有互相独立的指令集，并还包括主动控制的第二处理器及受控的第一处理器，所述第一处理器具有实际上比所述第二处理器的数据路径宽度要宽的数据路径宽度，包括以下步骤：

使所述不对称处理器中的受控的第一处理器运行；

使所述不对称处理器中的主动控制的第二处理器运行；

检测在所述第一处理器中的异常条件；

把指出所述第一处理器中的异常条件的中断请求信号从所述第一处理器发到所述第二处理器；

响应所述异常条件在所述第一处理器中引发空闲状态；以及

通过定义的方法来运行所述第二处理器以处理所述第一处理器的所述异常条件。

11. 按照权利要求 10 所述的方法，进一步包括以下步骤：

检测所述计算机系统的复位；以及

响应所述的计算机系统复位引发所述受控处理器的空闲状态。

12. 按照权利要求 11 所述的方法，其特征在于所述运行所述主动控制处理器以处理所述受控处理器的所述异常条件的步骤还包括以下步骤：

运行所述主动控制处理器以在所述受控处理器处于所述空闲状态时响应来自所述受控处理器的所述中断请求信号而初始化所述受控处理器的诸寄存器；以及

运行所述主动控制处理器以在所述受控处理器的所述诸寄存器初始化后执行把所述受控处理器激活成运行状态的指令。

13. 按照权利要求 12 所述的方法，其特征在于所述运行所述主动控制处理器以处理所述受控处理器的所述异常条件的步骤还包括以下步骤：

运行所述主动控制处理器以从所述受控处理器读取寄存器信息；

根据所述寄存器信息确定异常类型；以及

运行所述主动控制处理器以根据所述异常类型处理所述异常。

14. 按照权利要求 10 所述的方法，进一步包括以下步骤：

运行所述主动控制处理器以在所述受控处理器处于所述空闲状态时读和写所述受控处理器的各个寄存器。

15. 按照权利要求 10 所述的方法，进一步包括以下步骤：

使用单一的操作系统运行所述主动控制处理器和所述受控处理器；以及

在所述主动控制处理器和所述受控处理器中执行互相独立的指令集。

16. 按照权利要求 10 所述的方法，其特征在于：

所述受控处理器是具有大机器状态和大数据宽度的向量处理器或数据处理器；以及

所述主动控制处理器是具有比所述受控处理器的所述机器状态和数据宽度小得多的机器状态和数据宽度的控制处理器。

17. 一种多媒体计算机系统，包括：

控制处理器；

连接到所述控制处理器的向量或数据处理器，其中所述向量或数据处理器与所述控制处理器相比具有大机器状态和大数据宽度；

连接在所述控制处理器和所述向量或数据处理器之间的协处理器接口；

连接到所述向量或数据处理器的异常检测器，用于检测异常条件以及响应所述异常条件的检测向所述控制处理器发送中断请求及将所述向量或数据处理器去激活到空闲状态；以及

连接到所述控制处理器的异常处理器，用于接收所述中断请求及响应所述中断请求而处理所述向量或数据处理器的所述异常条件。

18. 按照权利要求 17 所述的多媒体计算机系统，其特征在于所述协处理器接口包括：

多个由所述控制处理器和所述向量或数据处理器可读和可写的寄存器；

译码器，用于对所述控制处理器指令集的指令译码并检测在所述协处理器接口上可执行的诸扩充指令；以及

执行所述诸扩充指令的状态机。

19. 按照权利要求 17 所述的多媒体计算机系统，其特征在于：

所述控制处理器和所述向量或数据处理器具有互不相同的控制和数据处理特性。

20. 按照权利要求 19 所述的多媒体计算机系统，其特征在于：

所述控制处理器具有规定的数据路径宽度；以及

所述向量或数据处理器具有比所述控制处理器的所述数据路径宽度大得多的宽数据路径宽度。

21. 按照权利要求 17 所述的多媒体计算机系统，进一步包括：

连接到所述控制处理器的主处理器，其中所述主处理器包括控制所述控制处理器运行的程序代码。

在不对称多处理器架构中处理中断和异常的系统和方法

[0001] 本发明涉及一种含有具有不相同控制和数据处理特性的不对称处理器的多处理器系统。具体说，本发明涉及在不对称多处理器系统中处理中断和异常的系统和方法。

[0002] 通用的处理器一般都包括处理中断和异常的电路和结构。中断是一种事件，该事件使处理器停止执行当前的指令线程 (instruction thread) 而开始执行由中断处理器指定和激活的中断服务指令线程。当完成中断服务指令线程的执行后，就继续执行当前指令线程。中断的起因与当前指令线程中各指令的执行并无直接关系，在此意义上说，中断是外部事件。处理中断的方法通常是保存处理器的状态或现场、执行中断服务程序、然后复原处理器的状态或现场并继续被中断的执行线程。

[0003] 异常是一种由处理器执行或企图执行某指令所引起的事件，执行该指令与处理器的当前状态不一致。造成异常条件的事件例子有 :企图由处理器执行非法或未定义的指令，企图在未对准指令地址或数据地址时执行指令，保护出错，已设置的断点，被零除，及执行算术指令时的溢出条件。处理器通过执行异常处理程序的指令线程来响应异常条件。在已有技术中，处理异常的方法是执行一套经过选择的指令集，具体执行哪个指令取决于造成异常条件的特定条件类型。

[0004] 中断和异常处理固有的控制操作用通用处理器进行起来既迅速又容易，但采用大现场的向量处理器则难得多，所以把多处理器架构系统中使用的中断和异常处理机构用于具有不相同控制和数据处理特性的处理器的不对称多处理器系统通常是低效的。使用中断和异常来处理的多处理器系统的操作经常涉及各处理器的状态的保存，而这种保存包括对诸寄存器和当前正在接受处理的数据的保存。在诸如多处理器的现场交换之类的操作期间，具有许多大寄存器的处理器不容易保存和重新加载处理器的状态。

[0005] 对于含有大机器状态或大现场的不对称处理器，需要的是便于处理中断和异常的多处理器架构。

[0006] 按照本发明的一个方面，多处理器计算机系统包括称为不对称处理器的多个处理器，这些处理器具有互不相同的控制和数据处理特性。虽然各别处理器的指令集和另一些处理器的指令集互不相关，但是这些不对称处理器受单一操作系统的控制。该多处理器计算机系统采用中断和异常处理的多处理架构定义，该定义规定由具有大机器状态和大数据宽度的称作数据或向量处理器的处理器来检测异常但是把中断和异常处理操作委托给另一个具有小机器状态和小数据宽度的称作控制处理器的处理器。由于控制程序一般涉及对个别状态标志和指针的监视和控制，所以控制处理器的小机器状态和小数据宽度十分适合于执行中断和异常处理之类的操作系统程序。与此相反，大机器状态和大数据宽度的数据或向量处理器处理控制任务的效果差，所以把中断和异常处理委托给控制处理器是非常有利的。

[0007] 按照本发明的另一方面，多处理器计算机系统包括在复位和检测到异常时进入空闲状态的数据处理器。该数据处理器进入空闲状态是为了方便系统的设计和编程，及为了在系统复位时简化各处理器的同步。

[0008] 按照本发明的又一实施例，多处理器计算机系统包括读写数据处理器内的各控制

和状态寄存器的控制处理器。因此该控制处理器控制了执行操作系统或应用程序期间数据处理器的操作。控制处理器进行独立于数据处理器执行的对数据处理器的各控制和状态寄存器的访问，使同一控制和状态寄存器可以由控制处理器和数据处理器并行地访问。

[0009] 按照本发明的再一实施例，多处理器计算机系统包括一种架构定义，其中控制处理器被约束只在数据处理器处于空闲状态时才访问该数据处理器的包括各控制和状态寄存器在内的内部状态。这一约束显著地简化了控制处理器和数据处理器之间的程序运行的相互作用，保护了数据处理器不致进入不协调和无效状态。

[0010] 所述不对称多处理器架构和操作方法具有许多优点。优点之一是即使与采用相同多个处理器的对称架构相比，编程模型和数据处理器的实现也得到显著的简化。

[0011] 在附后的权利要求书中明确陈述了所述实施例的被认为是新颖的特点。然而，与结构和运行方法二者有关的本发明实施例参考以下描述和附图可得到最好的理解。

[0012] 图 1 是表示按照本发明实施例的多媒体多处理器系统的高层次概略方框图。

[0013] 图 2 是表示包含在图 1 多媒体多处理器系统的多媒体信号处理器的概略方框图。

[0014] 图 3 是表示多媒体多处理器系统中的控制处理器的概略方框图。

[0015] 图 4 是控制处理器的功能图。

[0016] 图 5 是表示图 2 多媒体信号处理器中的向量处理器的概略方框图。

[0017] 图 6 是表示图 5 向量处理器的向量处理器执行数据路径的概略方框图。

[0018] 图 7 是表示图 2 多媒体信号处理器中的协处理器接口的概略方框图。

[0019] 图 8 是表示多媒体信号处理器的固体架构的概略方框图。

[0020] 图 9 是表示在多媒体多处理器系统中与中断和异常处理有关的操作的流程图。

[0021] 参照图 1，高层次概略方框图示出了包括主处理器 102 和多媒体信号处理器 104 的多媒体多处理器系统 100。典型的主处理器 102 是诸如 Pentium™ 或 Pentium Pro™ 之类的 x86 处理器。主处理器 102 根据保持在系统存储器 116 内的指令和数据来执行程序。主处理器 102 通过如 PCI 总线之类的系统总线 106 与多媒体信号处理器 104 通信。多媒体信号处理器 104 与各种功能块接口，如音频和通信 CODEC108、视频 A/D 变换器 110、视频 D/A 变换器 112、和帧缓冲 SDRAM 存储器 114。

[0022] 参照图 2，概略方框图示出多媒体多处理器系统 100 内部的多媒体信号处理器 104。多媒体信号处理器 104 包括连接到多个多媒体接口的数字信号处理器 (DSP) 核 202。

[0023] DSP 核 202 是多媒体信号处理器 104 的计算机器，包括 RISC 控制处理器 204、向量处理器 206、高速缓冲存储器子系统 208、快速总线 (FBUS)210 及 I/O 总线 212。控制处理器 204 是由英国 ARM Limited 公司设计和制造的 32 位 ARM 7™ 控制处理器，它执行通用处理功能如实时操作系统操作、中断和异常处理、输入 / 输出设备管理、与主处理器 102 的通信等等。在一个实施例中，控制处理器 204 工作在 40MHz。控制处理器 204 通过协处理器接口 242 与向量处理器 206 对接。

[0024] 控制处理器 204 响应于异常（通常是在处理指令时出现的不正常条件）而进行异常处理，从而导致执行控制流动的改变。控制处理器 204 响应于七种异常，它们按照从高到低的优先权次序列出为：复位条件、中止（数据）条件、FIQ、IRQ、中止（预取）条件、以及未定义的指令陷阱或软件中断。

[0025] 参照图 3，概略方框图示出了由指令译码器和控制逻辑 302 控制的 ARM7 控制处理

器 204。控制处理器 204 通过写数据寄存器 304 及指令流水线和读数据寄存器 306 与高速缓冲存储器子系统 208 通信。控制处理器 204 包括地址寄存器 308 和地址增量器 310 以对 31×32 位寄存器组 312 中的数据寻址。控制处理器 204 包括算术逻辑电路，如 32 位 ALU 314、桶形移位器 316 和 Booth 乘法器 318。协处理器接口 242 通过 nOPC、nCPI、CPA 和 CPB 信号线直接耦合到指令译码器和控制逻辑 302，这些信号线通过协处理器接口 242 在控制处理器 204 和向量处理器 206 之间交流操作码和指令变元。图 4 表示控制处理器 204 的功能图。

[0026] 向量处理器 206 是多媒体信号处理器 104 的数字信号处理机器。向量处理器 206 有单指令多数据的架构，并包括流水线化的 RISC 机器，该流水线化的 RISC 机器并行地对多个数据元素操作以执行如离散余弦变换 (DCT)、FIR 滤波、卷积、视频运动估计和其他处理操作之类的信号处理功能。向量处理器 206 支持向量算术，在该向量算术中以向量处理的方式用多个向量执行单元并行地对多个数据元素操作。向量处理器 206 执行标量操作，也执行组合的向量 - 标量操作。向量处理器 206 的多个数据元素打包成 576 位的向量，以每周 (例如，12.5ns) 三十二次 8/9 位定点算术操作、十六次 16 位定点算术操作或八次 32 位定点或浮点算术操作的速率进行计算。大多数 32 位标量操作的流水线速度为每周一个指令，而大多数 576 位向量操作的流水线速度为每二周一个指令。装入和存储操作与算术操作交叠并独立地由分别的装入和存储电路执行。

[0027] 参照图 5，向量处理器 206 有四个功能块，它们是取指令单元 502，指令译码器和发出器 504，指令执行数据路径 506，及装入和存储单元 508。取指令单元 502 及指令译码器和发出器 504 包含在向量处理器 206 内以允许向量处理器 206 的操作独立于控制处理器 204。

[0028] 取指令单元 502 预取指令并处理诸如分支指令和转移至子程序指令之类的控制流动指令。取指令单元 502 包括当前执行流用的预取指令的 16 项队列和分支目的流用的预取指令的 8 项队列。取指令单元 502 在一周内自指令高速缓冲存储器接收多达 8 条指令。指令译码器和发出器 504 译码和调度所有由向量处理器 206 执行的指令。译码器按照从取指令单元 502 接收的次序在一周内处理一条指令，而发出器根据执行资源和操作数据的可得情况不按次序地调度大多数指令。

[0029] 参照图 6，指令执行数据路径 506 包括四端口寄存器组 602，八个 32×32 并行乘法器 604 和八个 36 位 ALU 606。寄存器组 602 每个周期支持二次读操作和二次写操作。并行乘法器 604 每周产生整数或浮点格式的多达八次 32 位乘法，或十六次 16 位乘法或三十二次 8 位乘法。ALU 606 每周 (例如，12.5ns) 执行整数或浮点格式的八次 36 位 ALU 操作、十六次 16 位 ALU 操作或三十二次 8 位 ALU 操作。

[0030] 寄存器组 602 包括多个特殊用途寄存器和多个返回地址寄存器。特殊用途寄存器包括向量控制和状态寄存器 (VCSR)、向量程序计数器 (VPC)、向量异常程序计数器 (VEPC)、向量中断源寄存器 (VISRC)、向量和控制处理器同步寄存器 (VASYNC) 以及其他如各种计数、屏蔽、溢出和断点寄存器之类的寄存器。向量程序计数器 (VPC) 的计数值是待由向量处理器 206 执行的下一指令的地址。

[0031] 向量中断源寄存器 (VISRC) 向控制处理器 204 指示各中断源。一旦检测到异常，就由硬件设置 VISRC 的适当位。在向量处理器 206 恢复执行之前，用软件来使这些位复位。VISRC 中的任何一位都使向量处理器 206 进入 VP_IDLE(向量处理器 _ 空闲) 状态。如果

协处理器接口 242 的 VIMSK 寄存器中的对应中断允许位被置位，就向控制处理器 204 发出 IRQ 中断请求信号。

[0032] 向量处理器 206 检测包括精确异常和不精确异常在内的异常条件。精确异常由向量处理器 206 检测并在出错指令之前报告。精确异常包括指令地址断点异常、数据地址断点异常、无效指令异常、单步异常、返回地址堆栈溢出异常、返回地址堆栈下溢异常、VCINT 异常和 VCJOIN 异常。向量处理器 206 的不精确异常在执行按程序次序晚于出错指令的可变数目的指令之后检测和报告。不精确异常包括无效指令地址异常、无效数据地址异常、未对准的数据访问异常、整数溢出异常、浮点数溢出异常、浮点无效操作数据异常、浮点数除零异常及整数除零异常。

[0033] 当执行指令以中断控制处理器 204 时，向量中断指令寄存器 (VIINS) 用向量处理器的 VCINT 或 VCJOIN 指令更新。

[0034] 向量处理器 206 认得二个特别条件，它们是由控制处理器 204 执行的协处理器中断 (CPINT) 指令，和在向量处理器 206 中由软件执行的多个嵌套的转移运动子程序指令所引起的硬件堆栈溢出。在多媒体信号处理器 104 中产生的其他中断和异常条件由控制处理器 204 处理。

[0035] 一旦系统复位使向量处理器 206 处于空闲 (VP_IDLE) 状态，向量处理器 206 执行一串上电指令。向量处理器 206 保持在 VP_IDLE 状态直到控制处理器 204 用控制处理器指令集中的 STARTVP 指令初始化向量处理器 206 中的诸寄存器和激活向量处理器 206 的诸操作。

[0036] 再次参照图 2，控制处理器 204 初始化向量处理器 206 的诸操作。具体说，通过对控制处理器 204 指令集的扩充，控制处理器 204 控制与向量处理器 206 的交互作用。指令集的扩充包括协处理器数据操作（如 STARTVP 和 INTVP 指令）、协处理器数据传送、以及协处理器寄存器传送（如用于读写向量处理器 206 中寄存器的 TESTSET 指令和 MFVP、MTVP、MFER 和 MTER 指令）。

[0037] 控制处理器 204 用 STARTVP 指令起动向量处理器 206。STARTVP 指令使向量处理器 206 进入 VP_RUN 状态，表示向量处理器 206 正在执行中。如果向量处理器 206 已经处于 VP_RUN 状态，则 STARTVP 什么也不做。响应于 STARTVP 指令，没有结果传送给控制处理器 204，在 STARTVP 之后控制处理器 204 继续执行。在设置条件码的协处理器数据操作格式 (CDP) 指令之后引发 STARTVP 指令。STARTVP 指令用下列汇编器句法来引发：

[0038] CDP{cond} p7,0,c0,c0,c0

[0039] STARTVP{cond}

[0040] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 及其他条件。STARTVP 只在条件为真时执行。控制处理器 204 使用 STARTVP 指令向向量处理器 206 发出开始执行的请求信号，然后自动清除向量中断源 (VISRC) 寄存器的向量处理器中断位 (VISRC<vip>) 和向量处理器联合位 (VISRC<vjp>)。一旦作出 STARTVP 调用，控制处理器 204 不等待向量处理器 206 的开始执行就继续执行下一条指令。在调用 STARTVP 之前必须按需要初始化向量处理器 206 的状态。如果向量处理器 206 已经处于 VP_RUN 状态，则 STARTVP 指令不起作用。在 STARTVP 指令执行期间，可以出现向量处理器不可用的异常。

[0041] 控制处理器 204 使用中断向量处理器 (INTVP) 指令来停止向量处理器 206。INTVP 指令使向量处理器 206 进入表示向量处理器 206 不在执行的 VP_IDLE 状态。如果向量处理器 206 不在执行中，INTVP 什么事也不做。响应于 INTVP 指令，没有结果传送给控制处理器 204，在 INTVP 之后控制处理器 204 继续执行。在设置条件码的协处理器数据操作格式 (CDP) 指令之后引发 INTVP 指令。INTVP 指令只在条件为真时执行。因此，INTVP 指令用下列汇编器句法来引发：

[0042] CDP{cond} p7,1,c0,c0,c0

[0043] INTVP{cond}

[0044] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 及其他条件。控制处理器 204 使用 INTVP 指令向向量处理器 206 发出暂停执行的请求信号。一旦作出 INTVP 调用，控制处理器 204 不等待向量处理器 206 的暂停就继续执行下一条指令。一般在 INTVP 指令之后使用从扩充的寄存器传送 (MFER) 指令以确定向量处理器 206 是否已暂停。如果向量处理器 206 已经处于 VP_IDLE 状态，则 INTVP 指令不起作用。在 INTVP 指令执行期间可以出现向量处理器不可用的异常。

[0045] 控制处理器 204 使用测试和置位 (TESTSET) 指令来测试包括同步测试在内的向量处理器 206 的操作状态。TESTSET 读取为用户扩充的寄存器并把该寄存器的位 30 置为 1，从而提供了向量处理器 206 和控制处理器 204 之间的生产者 / 消费者类型的同步。TESTSET 使控制处理器 204 停车直至寄存器被传送。控制处理器 204 在设置条件码的协处理器寄存器传送操作 (MRC，MCR) 指令之后请求 TESTSET 指令。TESTSET 指令只在条件为真时执行。使用下列汇编器句法来调用 TESTSET 指令。

[0046] MRC{cond} p7,0,Rd,c0,cER,0

[0047] TESTSET{cond} Rd,RNAME

[0048] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 和其他条件。Rd 是控制处理器 204 中的寄存器。ER 是在协处理器接口 242 中的扩充的寄存器，RNAME 是指由架构规定的寄存器助记符如 UER1 或 VASYNC。

[0049] 其他控制处理器指令向控制处理器寄存器传送向量处理器标量寄存器 / 特殊用途寄存器中的数据。另一些其他的控制处理器指令把数据从控制处理器寄存器传送到向量处理器的标量 / 特殊用途寄存器。

[0050] 举例说，MFVP 指令将数据从向量处理器标量寄存器 / 特殊用途寄存器传送到控制处理器 204 中的通用寄存器。在设置条件码的协处理器寄存器传送操作 (MRC,MCR) 指令以后调用 MFVP 指令。MFVP 指令只在条件为真时执行。使用下列汇编器句法来调用 MFVP 指令：

[0051] MRC{cond} p7,1,Rd,CRn,CRm,0

[0052] MFVP{cond} Rd,RNAME

[0053] 其中条件 (cond) 包括等于 (eq)、不等于 (ne、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 和其他条件。协处理器号数码 p7 表示向量处理器 206。Rd 是控制处理器 204 中的寄存器。CRn 和 CRm 是向量处理器 206 中的标量寄存器 / 特殊用途寄存器。RNAME 是指由架构规定的寄存器助记符如 SP0 或 VCSR。控制处理器 204 调用 MFVP 指令以把数据从向量处理器 206 的标量寄存器 / 特殊用途寄存器 CRn 和 CRm 传送到控制处理器寄存器 Rd。

在执行 MFVP 指令期间可以出现向量处理器不可用的异常。

[0054] 向向量处理器传送 (MTVP) 指令把数据从控制处理器 204 中的通用寄存器传送到向量处理器标量寄存器 / 特殊用途寄存器。在设置条件码的协处理器寄存器传送操作 (MRC, MCR) 指令之后调用 MTVP 指令。MTVP 指令只在条件为真时执行。使用下列汇编器句法来调用 MTVP 指令 :

[0055] MRC{cond} p7,1,Rd,CRn,CRm,0

[0056] MTVP{cond} RNAME,Rd

[0057] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 及其他条件。协处理器号数码 p7 表示向量处理器 206。Rd 是控制处理器 204 中的寄存器。CRn 和 CRm 是向量处理器 206 中的标量寄存器 / 特殊用途寄存器。RNAME 是指由架构规定的寄存器助记符如 SP0 或 VCSR。控制处理器 204 调用 MTVP 指令以把控制处理器寄存器 Rd 中的数据传送到向量处理器 206 中的标量寄存器 / 特殊用途寄存器 CRn 和 CRm。在执行 MTVP 指令期间可以出现向量处理器不可用的异常。

[0058] MFVP 和 MTVP 只准备在向量处理器 206 处于 VP_IDLE 状态时执行。在控制处理器 204 和向量处理器 206 之间不发生直接通信。每次传送使用由协处理器接口 242 在示范性实施例中提供的中间存储器。

[0059] MFVP 和 MTVP 指令允许控制处理器 204 自由地读和写向量处理器 206 之中的各控制和状态寄存器。因此在执行操作系统或应用程序期间,控制处理器 204 控制向量处理器 206 的操作。MFVP 和 MTVP 指令只准备在向量处理器 206 处于 VP_IDLE 状态时执行,因为控制处理器 204 具有独立于向量处理器 206 的执行而访问向量处理器 206 的各控制和状态寄存器的能力,使同一的控制和状态寄存器可能被控制处理器 204 和向量处理器 206 并行地访问。

[0060] 向扩充的寄存器传送 (MTER) 指令把数据从控制处理器 204 传送到规定的扩充的寄存器。在设置条件码的协处理器寄存器传送操作 (MRC, MCR) 指令之后调用 MTER 指令。MTER 指令只在条件为真时执行。使用下列汇编器句法来调用 MTER 指令 :

[0061] MRC{cond} p7,1,Rd,cP,cER,0

[0062] MTER{cond} RNAME,Rd

[0063] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 和其他条件。协处理器号数码 p7 表示向量处理器 206。Rd 是控制处理器 204 中的寄存器。P 表示选择的协处理器,ER 表示在协处理器 P 中的扩充的寄存器。RNAME 是指由架构规定的寄存器助记符如 PER0 或 CSR。控制处理器 204 调用 MTER 指令以把控制处理器寄存器 Rd 中的数据传送到指定的协处理器 P 中指定的扩充的寄存器 ER。在执行 MTER 指令期间可以出现企图访问指定的协处理器时保护的破坏。

[0064] 从扩充的寄存器传送 (MFER) 指令把数据从指定的协处理器中的扩充的寄存器传送到控制处理器 204 中的寄存器。在设置条件码的协处理器寄存器传送操作 (MRC,MCR) 指令之后调用 MFER 指令。MFER 只在条件为真时执行。使用下列汇编器句法来调用 MFER 指令 :

[0065] MRC{cond} p7,2,Rd,cP,cER,0

[0066] MFER{cond} Rd,RNAME

[0067] 其中条件 (cond) 包括等于 (eq)、不等于 (ne)、进位置位 (cs)、进位清除 (cc)、负 (mi)、正 (pl) 和其他条件。协处理器号数码 p7 表示向量处理器 206。Rd 是控制处理器 204 中的寄存器。P 表示选择的协处理器，ER 表示在协处理器 P 中的扩充的寄存器。RNAME 是指由架构规定的寄存器助记符如 PER0 或 CSR。控制处理器 204 调用 MFER 指令以把数据从指定的协处理器 P 中指定的扩充的寄存器 ER 传送到寄存器 Rd。在执行 MFER 指令期间可以发生企图访问指定的协处理器时保护的破坏。

[0068] 协处理器接口 242 的概略方框图示于图 7。协处理器接口 242 通过增加实现扩充控制处理器 204 指令集的指令用的寄存器和逻辑功能来补充控制处理器 204 的功能度。协处理器接口 242 包括在控制处理器 204 和向量处理器 206 之间通信用的寄存器。协处理器接口 242 还充当在以不同时钟速率操作的结构之间交流数据和信号用的缓冲器。在一个实施例中，控制处理器 204 以 40MHz 速率操作，而向量处理器 206 以 80MHz 执行。

[0069] 协处理器接口 242 包括译码器 702、状态机 704、特许的扩充的寄存器块 706 和为用户们扩充的寄存器块 708。特许的扩充的寄存器块 706 中的各寄存器和用户扩充寄存器块 708 都是由控制处理器 204 和由向量处理器 206 可读和可写的。译码器 702 对控制处理器指令集的指令译码。控制处理器指令集包括可在控制处理器 204 上执行的指令以及不用控制处理器 204 执行而用协处理器接口 242 执行以实现协处理器特殊功能度的扩充指令。译码器 702 对控制处理器指令译码、检测扩充的指令及把检测到的扩充指令供应给状态机 704 以便执行。状态机 704 包括实现扩充指令用的逻辑电路。

[0070] 特许的扩充寄存器块 706 是在控制处理器 204 执行特殊指令期间被访问的扩充寄存器。特许的扩充寄存器块 706 包括控制寄存器 (CTR)、处理器版本寄存器 (PVR)、向量中断屏蔽寄存器 (VIMSK)、指令地址断点寄存器 (AIABR)、数据地址断点寄存器 (ADABR)、高速暂存寄存器 (SPREG) 和状态寄存器 (STR)。特许的扩充寄存器块 706 主要用于控制处理器 204 和向量处理器 206 的同步。

[0071] 在一个实施例中，为用户们扩充的寄存器块 708 的寄存器只有映射到位 <30> 的单一位，所以位 <31> 及位 <29:0> 都读作 0。为用户们扩充的寄存器块 708 包括向量处理器状态 (VPSTATE) 标志及向量和控制处理器同步 (VASYNC) 标志。VPSTATE 表示当位 <30> 置位时向量处理器 206 处于 VP_RUN 状态。当位 <30> 复位时，向量处理器 206 处于 VP_IDLE 状态并被暂停，此时程序计数器的地址指向下一条要执行的指令。VASYNC 允许向量处理器 206 和控制处理器 204 之间的生产者 - 消费者同步。VASYNC 标志由在向量处理器 206 中执行的 VMOV 指令来置位或复位。VASYNC 也通过使用从扩充的寄存器传送 (MFER) 指令、向扩充的寄存器传送 (MTER) 指令或 TESTSET 指令而由控制处理器 204 处理来置位或复位。

[0072] 一旦上电复位，特许的扩充寄存器块 706 和为用户们扩充的寄存器块 708 的所有各位都被复位，使得向量处理器 206 为空闲。

[0073] 向量中断屏蔽寄存器 (VIMSK) 控制向控制处理器 204 报告在向量处理器 206 内出现的异常。当 VIMSK 中的各位随向量中断源 (VISRC) 寄存器中的一相应位一起置位时，VIMSK 中的各位使异常能中断控制处理器 204。VISRC 寄存器包括表示多个异常和中断的来源的多个位。VIMSK 寄存器的各位包括数据地址断点中断允许 (DABE)、指令地址断点中断允许 (IABE) 和单步中断允许 (SSTPE)。VIMSK 还控制浮点溢出 (FOVE)、浮点无效操作数 (FINVE) 和浮点数除以零 (FDIVE) 的各中断允许位以及整数溢出 (IOVE) 和整数除以零

(IDIVE) 的各中断允许位。VIMSK 还控制 VCINT 中断允许 (VIE)、VCJOIN 中断允许 (VJE) 和现场交换允许 (CSE)。

[0074] 向量处理器 206 通过向控制处理器 204 发送信号与控制处理器 204 相互作用。具体说，向量处理器 206 包括向控制处理器 204 提供读访问和写访问的逻辑电路。向量处理器 206 用 VCJOIN 指令和 VCINT 指令通过为用户扩充的寄存器间接向控制处理器 204 送出表示向量处理器 206 已执行同步指令的信号。向量处理器 206 还通过表示向量处理器 206 已经暂停执行和已进入 VP_IDLE 状态的中断请求直接向控制处理器 204 发信号。向量处理器 206 执行用于告知控制处理器 204 的二条指令。VCJOIN 指令 (VCJOIN n) 有条件地与控制处理器 204 联合并使向量处理器 206 暂停和进入 VP_IDLE 状态。在向量处理器 206 内的程序计数器 (未示出) 指向 VCJOIN 指令后下一条指令的地址。VCINT 指令 (VCINT n) 有条件地中断控制处理器 204，同时使向量处理器 206 暂停和进入 VP_IDLE 状态。向量处理器 206 内的程序计数器指向 VCINT 指令后的下一条指令的地址。VCINT 和 VCJOIN 指令是由向量处理器 206 执行的归入控制流动类别的指令。控制流动指令包括如分支、减量然后分支、转移、自子程序返回、现场交换和屏蔽 (barrier) 指令之类的各种条件指令。

[0075] 在向量处理器 206 上执行的多个指令，包括条件现场交换 (VCCS) 指令、条件中断 (VCINT) 指令和条件联合 (VCJOIN) 指令，用于中断和异常处理。

[0076] 条件中断 (VCINT) 指令，如果被使能，暂停向量处理器 206 的执行和中断控制处理器 204。VCINT 指令可以产生 VCINT 中断。条件中断 (VCINT) 指令的汇编器句法如下 :

[0077] VCINT.cond # ICODE

[0078] 其中 cond 是规定的条件码，#ICODE 是标记供控制处理器 204 使用的常数的 23 位数值。条件码 (cond) 从以下条件码中选择 : 无条件 (un)、小于 (lt)、等于 (eq)、小于或等于 (le)、大于 (gt)、不等于 (ne)、大于或等于 (ge) 及溢出 (ov)。VCINT 的操作由伪码描述如下 :

[0079] If((cond == VCSR[SO,GT,EQ,LT])|(cond == un)){

[0080] VISRC<vip> = 1 ;

[0081] VIINS = [VCINT.cond # ICODE instruction] ;

[0082] VEPC = VPC ;

[0083] if(VIMSK<vie> == 1)signal control processor int ;

[0084] VP_STATE = VP_IDLE ;

[0085] }

[0086] else VPC = VPC+4 ;

[0087] 因此，如果向量控制和状态寄存器 (VCSR) 与条件码相配或者条件码为无条件，则向量中断源 (VISRC) 寄存器的向量中断位置位，向量中断指令寄存器 (VIINS) 装入由 #ICODE 参数标记的指令，向量异常程序计数器 (VEPC) 装入来自向量处理器 206 的程序计数器的数值。向量中断指令寄存器 (VIINS) 装入 VCINT 指令的标记符、条件和 #ICODE 变元以向控制处理器 204 供给与以下有关的信息 : (1) 向量处理器 206 要中断控制处理器 204 中的原因和 (2) 在执行软件中断服务程序中供控制处理器 204 使用的常数值。如果 VIMSK 寄存器的向量处理器中断允许位被置位，则向量处理器 206 向控制处理器 204 发送中断信号 (IRQ)。向量处理器 206 的状态 (VP_STATE) 被设成空闲状态。然而，如果条件码不是无条

件或者控制码与向量控制和状态寄存器 (VCSR) 不配，则 VCINT 指令被忽视，但是程序计数器增四。

[0088] 有条件与控制处理器任务联合 (VCJOIN) 指令，如果被使能，暂停向量处理器 206 的执行并参加在控制处理器 204 中工作的任务。VCJOIN 指令可以产生 VCJOIN 中断。条件联合 (VCJOIN) 指令的汇编器句法如下：

[0089] VCJOIN.cond # Offset

[0090] 其中 cond 是规定的条件码，#Offset 是标记相对于向量处理器 206 存储器中 VCJOIN 指令位置的偏移量的 23 位数值，其中的存储器存有在以后待激活的向量处理器子程序。VCJOIN 指令激活在控制处理器 204 中执行的中断服务程序。控制处理器 204 保存 #Offset 参数并执行规定的中断服务程序。中断服务程序完成时，控制处理器 204 进行规定的操作并对向量处理器 206 中的诸寄存器置位以作将来的处理。一旦完成中断服务程序，控制处理器 204 用 STARTVP 指令起动向量处理器 206，并指定保存的地址 #Offset 作为向量处理器 206 应该开始执行的子程序的位置。控制处理器 204 以分支指令的方式使用偏移量变元以引导控制处理器 204 执行的指令序列。使用 VCJOIN 指令的一个例子描述于美国专利申请号 ××/×××，×××，代理人案号 M-4365US，S.P.Song 等人发明的标题为“EFFICIENT CONTEXT SAVING ANDRESTORING IN MULTIPROCESSORS(多处理器中有效的现场保存和恢复)”的申请，为了本文的完整将其纳入作为参考文献。条件码 (cond) 从以下的条件码中选择：无条件 (un)、小于 (lt)、等于 (eq)、小于或等于 (le)、大于 (gt)、不等于 (ne)、大于或等于 (ge) 和溢出 (ov)。VCJOIN 的操作由伪码描述如下：

```
[0091] If((cond == VCSR[SO,GT,EQ,LT])|(cond == un)){
[0092]       VISRC<vjp> = 1;
[0093]       VIINS = [VCJOIN.cond # Offset instruction];
[0094]       VEPC = VPC;
[0095]       if(VIMSK<vje> == 1)signal control processor
[0096] int;
[0097]       VP_STATE = VP_IDLE;
[0098]   }
[0099]   else VPC = VPC+4;
```

[0100] 因此，如果向量控制和状态寄存器 (VCSR) 与条件码相配或者条件码为无条件，则向量中断源 (VISRC) 寄存器的向量联合位被置位，向量中断指令寄存器 (VIINS) 装入由 #Offset 参数标记的指令，向量异常程序计数器 (VEPC) 装入来自向量处理器 206 的程序计数器的数值。向量中断指令寄存器 (VHNS) 装入 VCJOIN 指令的标记符、条件和 #Offset 变元以向控制处理器 204 供给与以下有关的信息：(1) 向量处理器 206 要中断控制处理器 204 的原因和 (2) 向量处理器 206 的指令存储器中自地址或标签 (label) 的偏移值，该偏移值表示在控制处理器 204 重新激活向量处理器 206 时引导向量处理器例行程序的指令顺序流动的分支偏移值。如果 VIMSK 寄存器的向量处理器联合允许位被置位，则向量处理器 206 向控制处理器 204 发送中断信号 (IRQ)。向量处理器 206 的状态 (VP_STATE) 被设成空闲状态。然而，如果条件码不是无条件或者控制码与向量控制和状态寄存器 (VCSR) 不配，则 VCJOIN 指令被忽视，但是程序计数器增四。

[0101] 如果 VIMSK 寄存器的现场交换允许位 VIMSK<cse> 为真，条件现场交换 (VCCS) 指令执行向现场交换子程序的立即（不延时的）转移。如果 VIMSK<cse> 为真，则把 VPC+4 中的返回地址保存到返回地址堆栈上。否则，在地址 VPC+4 处异常继续。VCCS 指令可以产生返回地址堆栈溢出异常。条件现场交换 (VCCS) 指令的汇编器句法如下：

[0102] VCCS # Offset

[0103] 其中 #Offset 是标记现场交换子程序位置的 23 位数值。VCCS 操作由伪码描述如下：

[0104] If VIMSK<cse> == 1){

[0105] if(VSP<4> > 15){

[0106] VISRC<RASO> = 1;

[0107] signal control processor with RASO;

[0108] VP_STATE = VP_IDLE;

[0109] } else {

[0110] RSTACK[VSP<3:0>] = VPC+4;

[0111] VSP<4:0> = VSP<4:0>+1;

[0112] VPC = VPC+4;

[0113] }

[0114] } else VPC = VPC+4;

[0115] 因此，如果 VIMSK 寄存器的现场交换允许位被置位，则测试返回堆栈指针 (VSP) 的溢出。如果已发生溢出条件，则向量中断源 (VISRC) 寄存器的返回地址堆栈溢出位被置位，向量处理器 206 向控制处理器 204 发送返回地址堆栈溢出异常，向量处理器 206 的状态 (VP_STATE) 被置成空闲状态。如果不曾发生溢出条件，则加四后的程序计数器值被压入返回地址堆栈 (RSTACK)，返回堆栈指针 (VSP) 被加一，向量处理器 206 的程序计数器 (VPC) 被增以 #Offset 乘四的带符号乘积。如果 VIMSK 寄存器的现场交换允许位没有置位，则忽视 VCCS 指令，但是程序计数器增四。

[0116] 一旦在控制处理器 204 执行 STARTVP 指令之后向量处理器 206 开始执行操作，控制处理器 204 和向量处理器 206 就互相独立地并行操作。控制处理器 204 和向量处理器 206 互相独立和并行的操作利用特殊控制指令同步。对二个线程（一个在控制处理器 204 上，另一个在向量处理器 206 上）同时执行的支持有利地增加了任务执行速度，还便于把在控制处理器 204 上运行的指令码移植到 MSP 处理器系列的种种处理器。在本说明性的实施例中，向量处理器 206 以 80MHz 的速率工作。

[0117] 再次参看图 2，高速缓冲存储器子系统 208 包括数据高速缓冲存储器 214（例如 5KB）、指令高速缓冲存储器 216（例如 2KB）和高速缓冲存储器 ROM218（例如 16KB），该子系统 208 一般工作在向量处理器 206 的同一速率 (80MHz) 上。在一个实施例中，高速缓冲存储器子系统 208 包括控制处理器 204 用的 1KB 指令存储器和 1KB 数据存储器、向量处理器 206 用的 1KB 指令存储器和 4KB 数据存储器以及控制处理器 204 和向量处理器 206 合用的 16KB 集成的指令和数据高速缓冲存储器 ROM。高速缓冲存储器子系统 208 通过 32 位数据总线与控制处理器 204 接口，通过 128 位数据总线与向量处理器 206 接口。高速缓冲存储器 ROM 218 包括 uROM 初始化软件、自检诊断软件、各种系统管理软件、库例行程序及用于经

过挑选的指令和数据常数的高速缓冲存储器。具体说，高速缓冲存储器 ROM 218 包括控制处理器 204 用的指令异常处理器和输入输出设备中断处理器 0、1、2 和 3。高速缓冲存储器 ROM218 也包括在控制处理器 204 中执行的向量处理器中断处理器和向量处理器断点异常处理器。

[0118] FBUS 210 与多个 FBUS 外设对接，这些 FBUS 外设包括例如 32 位 PCI 总线接口 220、64 位 SDRAM 存储器控制器 222、8 通道 DMA 控制器 224、客户 ASIC 逻辑块 226 及存储器数据传送器 228。PCI 总线接口 220 连接至系统总线 106 并工作于例如 33MHz。客户 ASIC 逻辑块 226 按需要提供实现常规功能的控制逻辑。在一个实施例中，客户 ASIC 逻辑块 226 提供包括与各种模拟 CODEC 和客户专用的 I/O 设备接口的 10K 个门。存储器数据传送器 228 把 DMA 数据从主处理器 102 传送给在多媒体信号处理器 104 内部的 SDRAM 存储器 230。

[0119] I/O 总线 212 与多个 I/O 总线设备对接，这些设备包括位流处理器 232、UART 串行线 234、定时器电路 236、中断控制器 238 和特殊寄存器 240。位流处理器 232 处理视频位流。特殊寄存器 240 用于受软件控制的初始化和中断处理。

[0120] 参照图 8，概略方框图示出了多媒体信号处理器 104 的软件和固件架构 800，包括在多媒体信号处理器 104 上执行的 MSP 系统部件软件 802 和在主处理器 102 上执行的 PC 应用程序和操作系统软件 808。多媒体信号处理器 104 受固件控制，该固件包括在向量处理器 206 上执行的向量化 -DSP 固件库 804 和在控制处理器 204 上执行的系统管理功能块 806。向量化 -DSP 固件库 804 和系统管理功能块 806 包括在 MSP 系统部件软件 802 中。架构 800 的优点是把信号处理的功能从主处理器应用控制操作中分离出来以简化软件的开发，它改进了软件设计管理和降低了应用程序开发和维护的成本。

[0121] MSP 系统部件软件 802 专门在控制处理器 204 上执行，它包括 MSP 实时核心 810、多媒体库模块 812、系统管理功能块 806 和向量化 -DSP 固件库 804。MSP 实时核心 810 一般负责与主处理器 102 的接口、资源管理、I/O 设备处理和大部分中断及异常处理。MSP 实时核心 810 包括与在主处理器 102 执行的与 Windows™ 和 WindowsNT™ 软件接口用的软件。MSP 实时核心 810 还包括从主处理器 102 选择和下载已选应用固件的软件、在控制处理器 204 和向量处理器 206 中执行用的调度诸任务的软件以及管理多媒体信号处理器 104 的包括存储器和 I/O 设备在内的系统资源的软件。MSP 实时核心 810 包括使多媒体信号处理器 104 各任务之间的通信同步的软件和报告与 MSP 有关的中断、异常和状态条件的软件。

[0122] 向量化 -DSP 固件库 804 实质上执行所有数字信号处理功能。向量化 -DSP 固件库 804 还控制由控制处理器 204 向向量处理器 206 发出的如协处理器中断之类的专用特殊中断，或控制在向量处理器 206 内产生的硬件堆栈溢出异常。

[0123] 多媒体库模块 812 执行通信处理功能如数据通信、MPEG 视频和音频、语言编码和合成、与 Sound Blaster™ 兼容的音频等。MSP 实时核心 810 是实时、健壮 (robust)、多任务、抢先的操作系统，它包括便于在多媒体信号处理器 104 上执行多媒体应用程序的增强措施。

[0124] 在主处理器 102 中执行的 PC 应用和操作系统软件 808 控制多媒体信号处理器 104，其方法是通过系统总线 106 读写各 MSP 控制和状态寄存器，及写入驻留到系统存储器 116 和驻留到多媒体信号处理器 104 的共用数据结构。

[0125] MSP 的程序执行随控制处理器 204 执行第一执行流而开始。控制处理器 204 可以

在向量处理器 206 起动第二个独立的执行流。控制处理器 204 和向量处理器 206 的操作用以下指令同步：在控制处理器 204 中运行的专用协处理器指令，包括 STARTVP，INTVP 和 TESTVP 指令；在向量处理器 206 中执行的特殊指令，包括 VJOIN 和 VINT 指令。控制处理器 204 和向量处理器 206 之间的数据传送用在控制处理器 204 中执行的数据传送指令来实现。

[0126] 在上电之后，多媒体信号处理器 104 自动进入自检顺序以充分测试功能性。自检顺序包括多媒体信号处理器 104 中所有寄存器的初始化和检验多媒体信号处理器 104 各部件功能性的片内自检诊断程序的执行。一旦自检顺序结束，多媒体信号处理器 104 就执行 MSP 系统部件软件 802 以装入和执行 MSP 初始化软件及装入和执行 MSP 实时核心 810。多媒体信号处理器 104 支持三种复位操作，它们是：通过 PCI 总线的受硬件控制的系统复位、通过 MSP 控制寄存器中 PCI 系统复位位的受软件控制的系统复位以及通过控制处理器 204 和通过 MPS 控制寄存器内向量重新起动位的受软件控制的重新起动。

[0127] 参照图 9，流程图示出与多媒体多处理器系统 100 中中断和异常处理有关的操作。

[0128] 向量处理器 206 在多媒体多处理器系统 100 中执行非常有限的控制和管理功能。具体说，虽然向量处理器 206 产生异常，向量处理器 206 却不包括处理异常的能力。相反，复位初始化和各异常由控制处理器 204 来处理。此外，只在向量处理器 206 空闲时才允许控制处理器 204 去读或写向量处理器 206。因此，控制处理器 204 作为主动控制的处理器工作，而向量处理器 206 作为在主动控制处理器控制下的受控处理器工作。

[0129] 如此，在向量处理器 206 和控制处理器 204 具有互不相同的控制和数据处理特性的意义上，多媒体多处理器系统 100 包括多个不对称的处理器。这些不对称处理器由单一的操作系统控制。向量处理器 206 和控制处理器 204 具有互不相关的指令集。多媒体多处理器系统 100 采用中断和异常处理的多处理器架构定义，其中向量处理器 206 具有大机器状态或现场、大数据宽度和许多寄存器。在一个说明性实施例中，向量处理器 206 具有 288 位数据宽度、64 个 288 位向量寄存器、80 个以上的 32 位标量寄存器和 4KB 高速暂存 (scratch pad) 存储器。向量处理器 206 的大机器状态包括具有相当大量位数的相当大量寄存器。向量处理器 206 对于阵列或向量之类的大数据结构的处理十分有效，但是对于位和状态标志之类的小数据结构的处理不是最优的。因此向量处理器 206 包括检测异常的电路，但是把中断和异常处理操作委托给控制处理器 204。控制处理器 204 具有小机器状态和小数据宽度，例如 16 或 32 位，这对执行中断和异常处理之类的操作系统程序是有利的，因为控制程序一般涉及个别状态标志和指针的监视和控制。

[0130] 向量处理器 206 检测异常条件但是不为该条件服务而是向控制处理器 204 发送中断信号、进入空闲状态并且保持在空闲状态直到引起异常的条件或来源接受了控制处理器 204 的操作的服务。由于异常和中断控制操作不需要协调，由于所有控制操作都单独由控制处理器 204 执行，所以简化了多媒体多处理器系统 100 的控制。

[0131] 向量处理器 206 工作在按照 MSP 架构定义的运行状态 (VP_RUN) 和空闲状态 (VP_IDLE) 以分别指示向量处理器 206 正在执行或是无限期暂停执行。一旦多媒体多处理器系统 100 复位 902，向量处理器 206 进入 VP_IDLE 状态 904。控制处理器 204 仅在 VP_IDLE 状态期间才被允许读写向量处理器 206 中的寄存器 906。当向量处理器 206 处于 VP_RUN 状态时，由控制处理器 204 执行的和指向向量处理器 206 的寄存器的读操作或写操作的结果是“有界无定义的 (boundedly undefined)”。“有界无定义的”结果是多媒体信号处理器 104

中任何指令序列都可产生的多媒体多处理器系统100的状态。因此，只允许控制处理器204在向量处理器206处于VP_IDLE状态时进行对向量处理器206内各寄存器的读写访问。

[0132] 控制处理器204初始化向量处理器206中的诸寄存器908。在把向量处理器206中的诸寄存器初始化到选定的状态之后，控制处理器204执行STARTVP指令910以把向量处理器206的状态改变为VP_RUN状态911。

[0133] 当向量处理器206遭遇异常条件912时，向量处理器206进入VP_IDLE状态914并用中断请求916告知控制处理器204。向量处理器206保持在VP_IDLE状态直到控制处理器204执行另一条STARTVP指令918。控制处理器204通过读出向量处理器206中的诸寄存器而确定所报告的异常的类别920并以规定的方式处理该异常922。然后控制处理器204，如果有需要，重新初始化924向量处理器206中的诸寄存器，接着重新起动向量处理器206的执行926。

[0134] 在保存和恢复向量处理器206中的诸寄存器方面，向量处理器206比控制处理器204更有效。因此，当控制处理器204编程使向量处理器206在现场交换操作期间执行寄存器保存和恢复指令时，实现了性能上的优点。一种有利的现场交换操作的详细讨论见于在本发明同一申请日申请的、一起正在审理中的、美国专利申请号××/×××，×××（代理人案号M-4365）、S.P.Song等人发明的标题为“EFFICIENT CONTEXT SAVING ANDRESTORING IN MULTIPROCESSORS（多处理器中有效的现场保存和恢复）”的申请。

[0135] 当复位和检测到异常时，向量处理器206进入VP_IDLE状态以方便系统的设计和编程及在系统复位时简化处理器的同步。异常一般不经常出现，所以在不显著降低向量处理器206功能度或性能的情况下大大简化了向量处理器206的中断处理。此外，普遍认为对异常的响应性是重要性有限的特性。例如，许多实时执行的现有应用程序诸如多媒体应用程序包括在实时操作模式下禁止对异常事件作出响应的编程。在另一例中，某些现有的应用程序使用饱和模式（saturation mode）来避免实时算术处理时的算术或下溢条件。

[0136] 虽然参照各种实施例对本发明作了描述，但是应当懂得这些实施例是说明性的，本发明的范围不受它们的限制。可能对所述的实施例作出许多变动、修改、添加和改进。例如，这些实施例是作为利用包括Pentium主计算机和特定多媒体处理器的多处理器系统的系统来描述的。在其他实施例中可以采用其他的处理器配置。

[0137] 本专利文件与以下同时申请的专利有关，为了完整起见，将它们纳入作为参考：

[0138] 美国专利申请号××/×××，×××，代理人案号M-4354US，标题“MULTIPROCESSOR OPERATION IN A MULTIMEDIA SIGNALPROCESSOR（在多媒体信号处理器中的多处理器操作）”，发明者Le Nguyen。

[0139] 美国专利申请号××/×××，×××，代理人案号M-4355US，标题“SINGLE-INSTRUCTION-MULTIPLE-DATA PROCESSING IN AMULTMEDIA SIGNAL PROCESSOR（在多媒体信号处理器中的单指令多数据处理）”，发明者Le Nguyen。

[0140] 美国专利申请号××/×××，×××，代理人案号M-4365US，标题“EFFICIENT CONTEXT SAVING AND RESTORING IN MULTIPROCESSORS（多处理器中有效的现场保存和恢复）”，发明者S.P.Song等人。

[0141] 美国专利申请号××/×××，×××，代理人案号M-4366US，标题“SYSTEM AND METHOD FOR HANDLING SOFTWARE INTERRUPTSWITH ARGUMENT PASSING（随变元传送处理软

件中断的系统和方法）”，发明者 S.P.Song 等人。

[0142] 美国专利申请号 ××/×××，×××，代理人案号 M-4368US，标题“METHODS AND APPARATUS FOR PROCESSING VIDEO DATA（处理视频数据的方法和装置）”，发明者 C.Reader 等人。

[0143] 美国专利申请号 ××/×××，×××，代理人案号 M-4369US，标题“SINGLE-INSTRUCTION-MULTIPLE-DATAPROCESSING USINGMLTIPLE BANKS OF VECTOR REGISTERS（使用多组向量寄存器的单指令多数据处理）”，发明者 Le Nguyen 等人。

[0144] 美国专利申请号 ××/×××，×××，代理人案号 M-4370US，标题“SINGLE-INSTRUCTION-MULTIPLE-DATAPROCESSING WITHCOMBINED SCALAR/VECTOR OPERATIONS（带标量/向量组合操作的单指令多数据处理）”，发明者 Le Nguyen 等人。

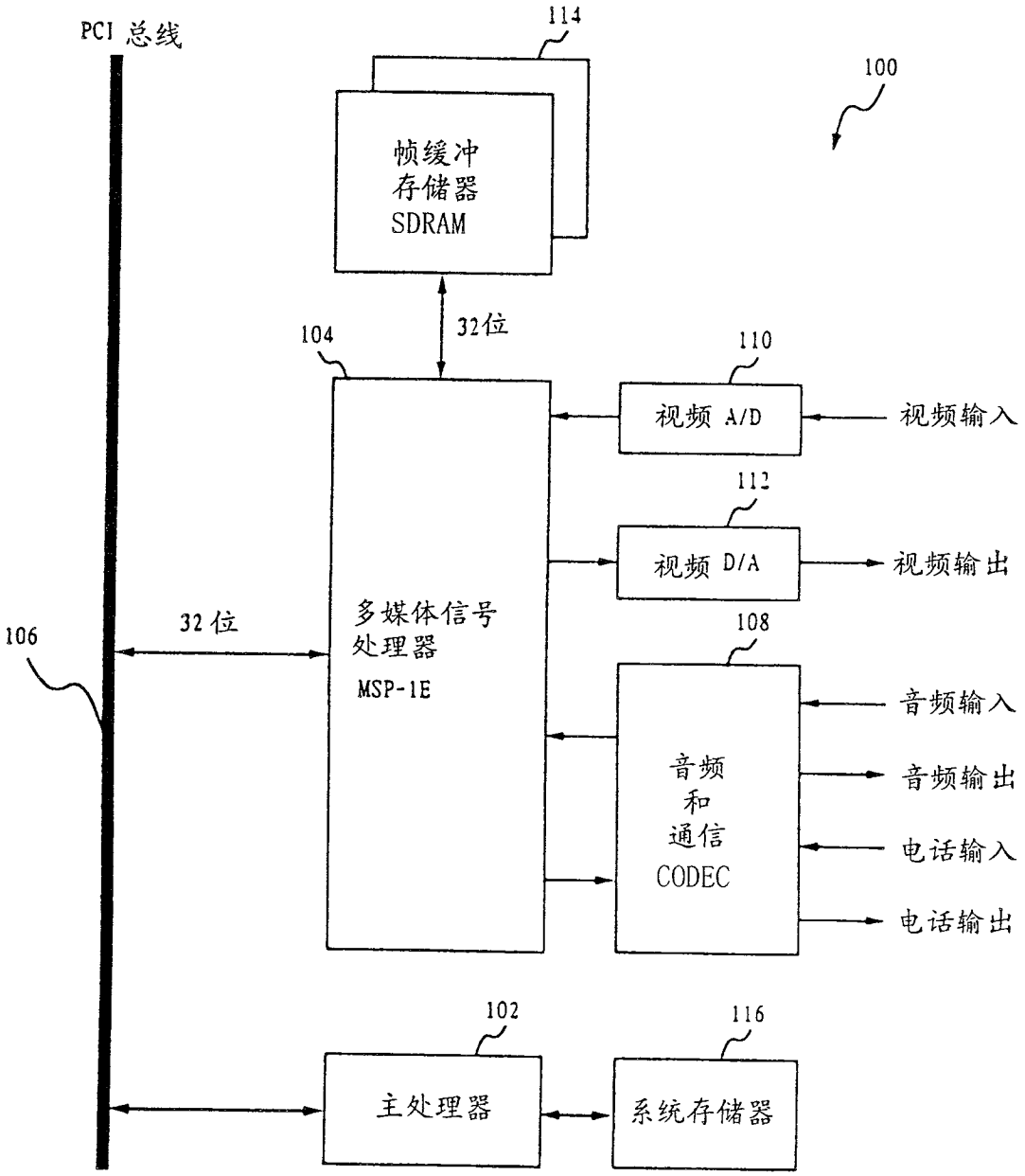

PCI 总线
114
帧缓冲
存储器
SDRAM
100
32位
104
110
视频 A/D
视频输入
112
视频 D/A
视频输出
106
32 位
多媒体信号
处理器
MSP-1E
108
音频输入
音频输出
音频
和
通信
CODEC
电话输入
电话输出
102
116
主处理器
系统存储器

图 1

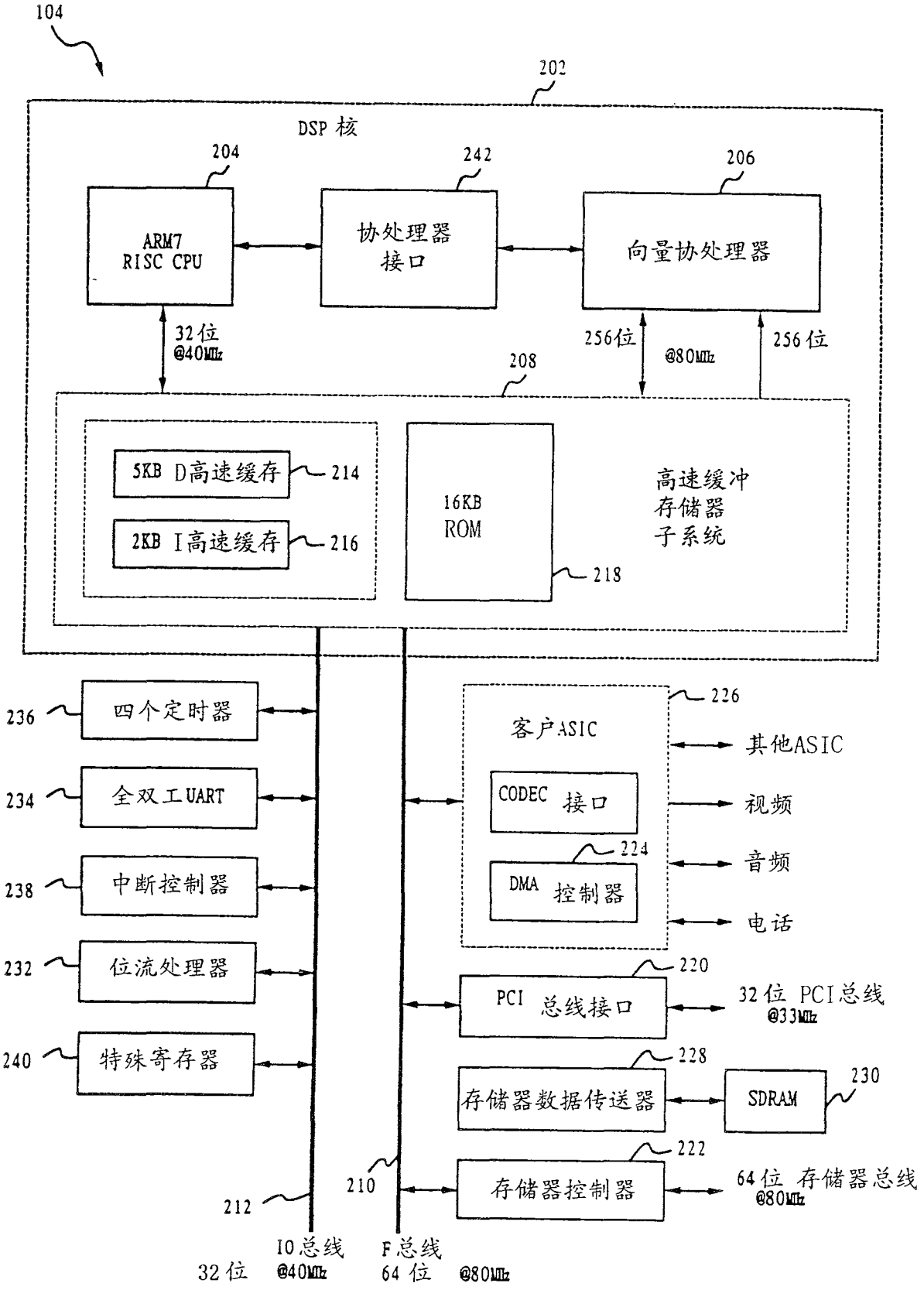

104
202
DSP 核
204
ARM7
RISC CPU
242
协处理器
接口
206
向量协处理器
32位
@40MHz
256位
@80MHz
256 位
208
5KB D高速缓存
214
2KB I高速缓存
216
16KB
ROM
218
高速缓冲
存储器
子系统
236
四个定时器
234
全双工UART
238
中断控制器
232
位流处理器
240
特殊寄存器
226
客户ASIC
其他ASIC
CODEC 接口
视频
224
DMA 控制器
音频
电话
220
PCI 总线接口
32 位 PCI总线
@33MHz
228
存储器数据传送器
SDRAM
230
222
存储器控制器
64 位 存储器总线
@80MHz
210
212
IO总线
32 位 @40MHz
F总线
64 位 @80MHz

图 2

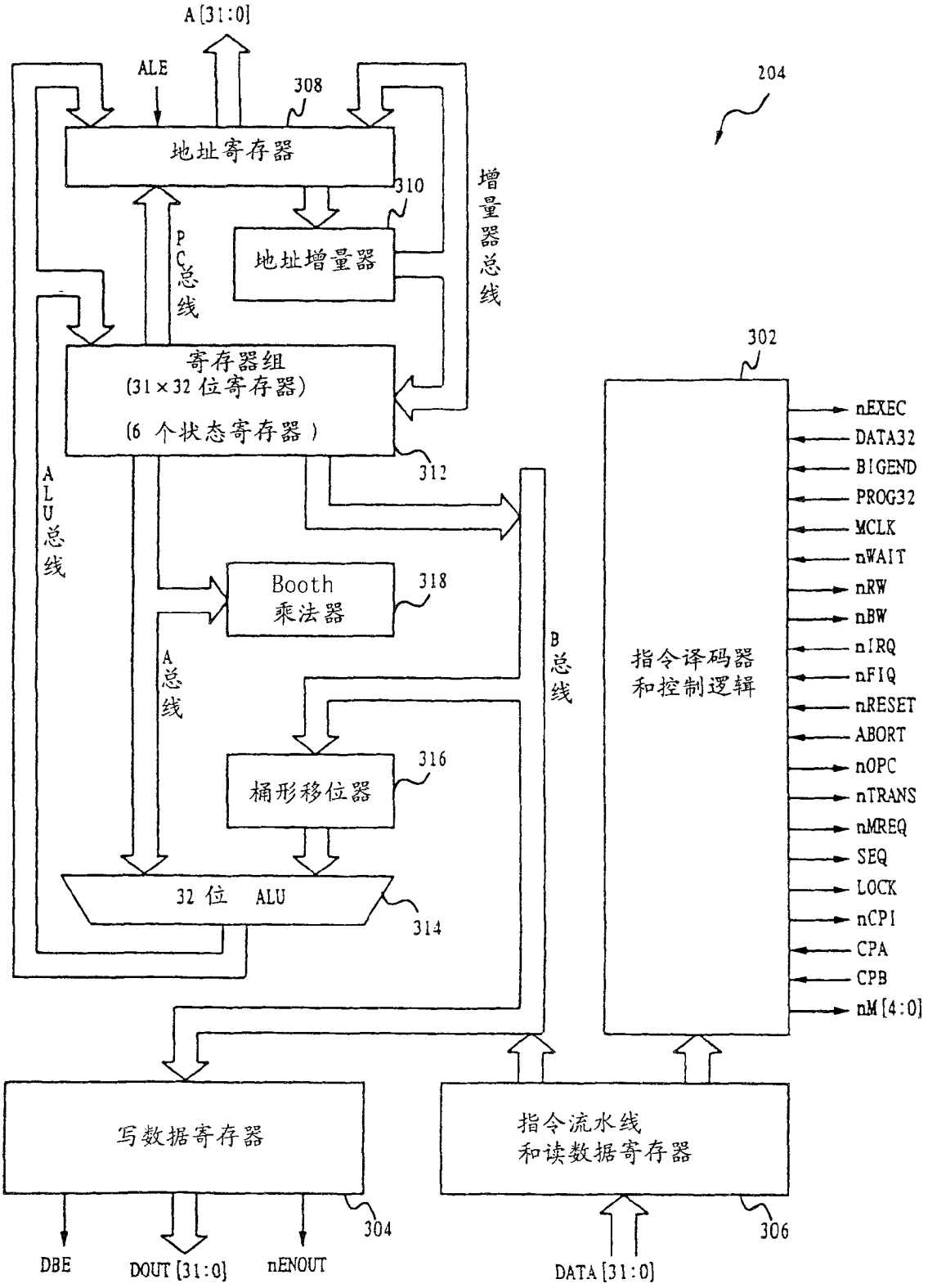

A[31:0]
ALE
308
204
地址寄存器
310
地址增量器
增量器总线
PC总线
寄存器组
(31×32 位寄存器)
(6 个状态寄存器)
312
302
ALU总线
Booth
乘法器
318
A总线
B总线
指令译码器
和控制逻辑
nEXEC
DATA32
BIGEND
PROG32
MCLK
nWAIT
nRW
nBW
nIRQ
nFIQ
nRESET
ABORT
nOPC
nTRANS
nMREQ
SEQ
LOCK
nCPI
CPA
CPB
nM[4:0]
316
桶形移位器
32 位 ALU
314
写数据寄存器
304
指令流水线
和读数据寄存器
306
DBE
DOUT[31:0]
nENOUT
DATA[31:0]

图 3

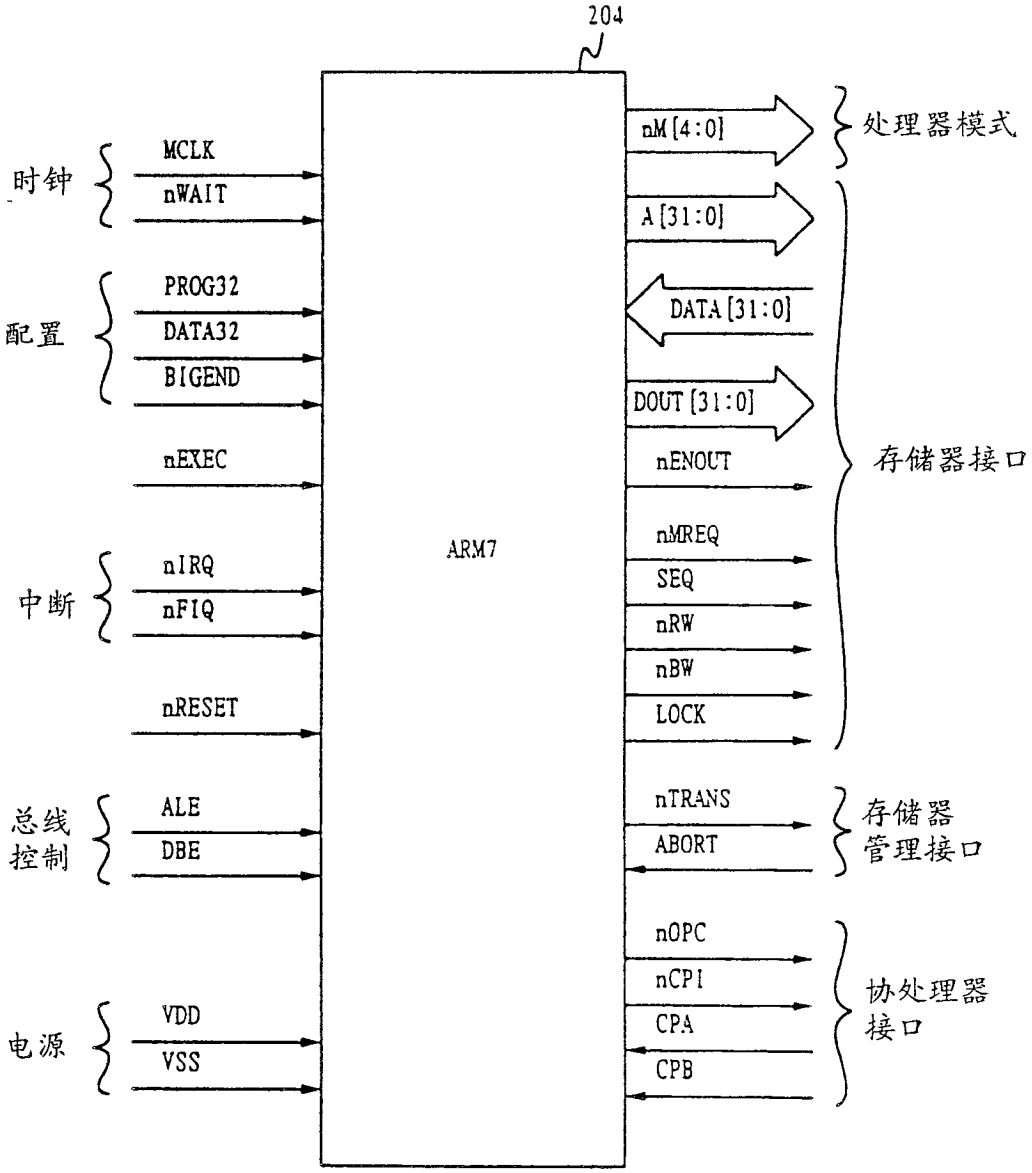

204
ARM7
时钟
MCLK
nWAIT
配置
PROG32
DATA32
BIGEND
nEXEC
中断
nIRQ
nFIQ
nRESET
总线控制
ALE
DBE
电源
VDD
VSS
nM[4:0]
处理器模式
A[31:0]
DATA[31:0]
DOUT[31:0]
nENOUT
nMREQ
SEQ
nRW
nBW
LOCK
存储器接口
nTRANS
ABORT
存储器管理接口
nOPC
nCPI
CPA
CPB
协处理器接口

图 4

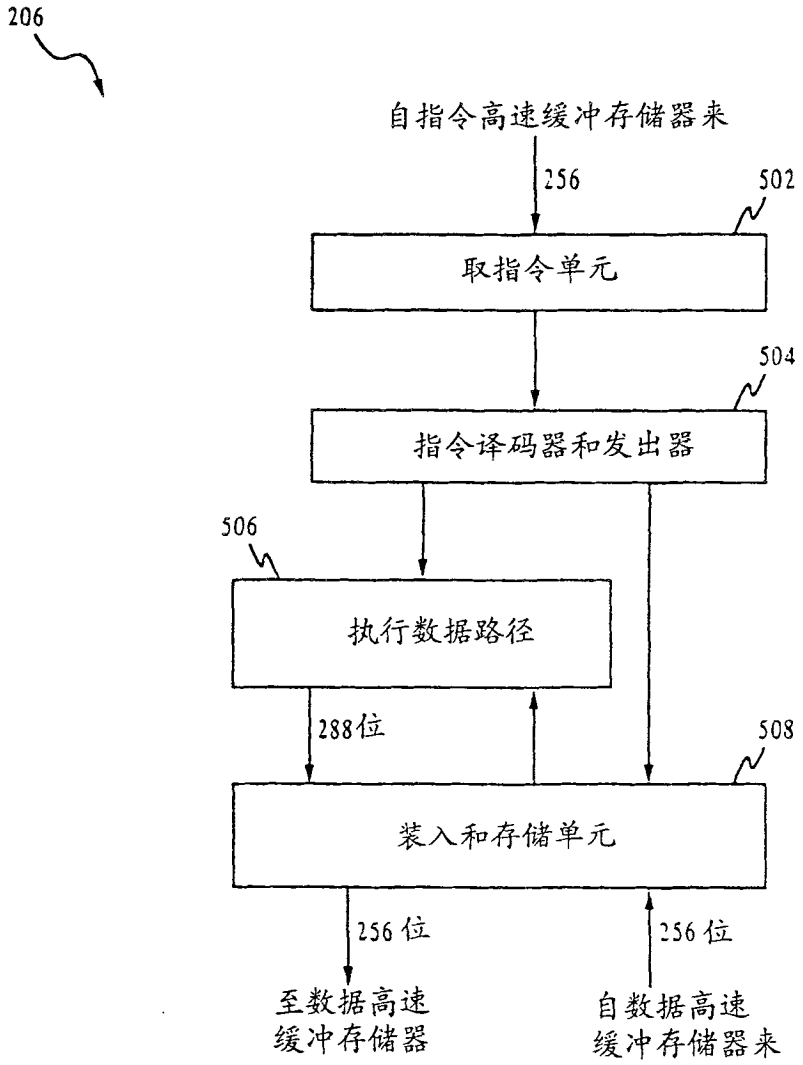

206
自指令高速缓冲存储器来
256
502
取指令单元
504
指令译码器和发出器
506
执行数据路径
288位
508
装入和存储单元
256 位
256 位
至数据高速
缓冲存储器
自数据高速
缓冲存储器来

图 5

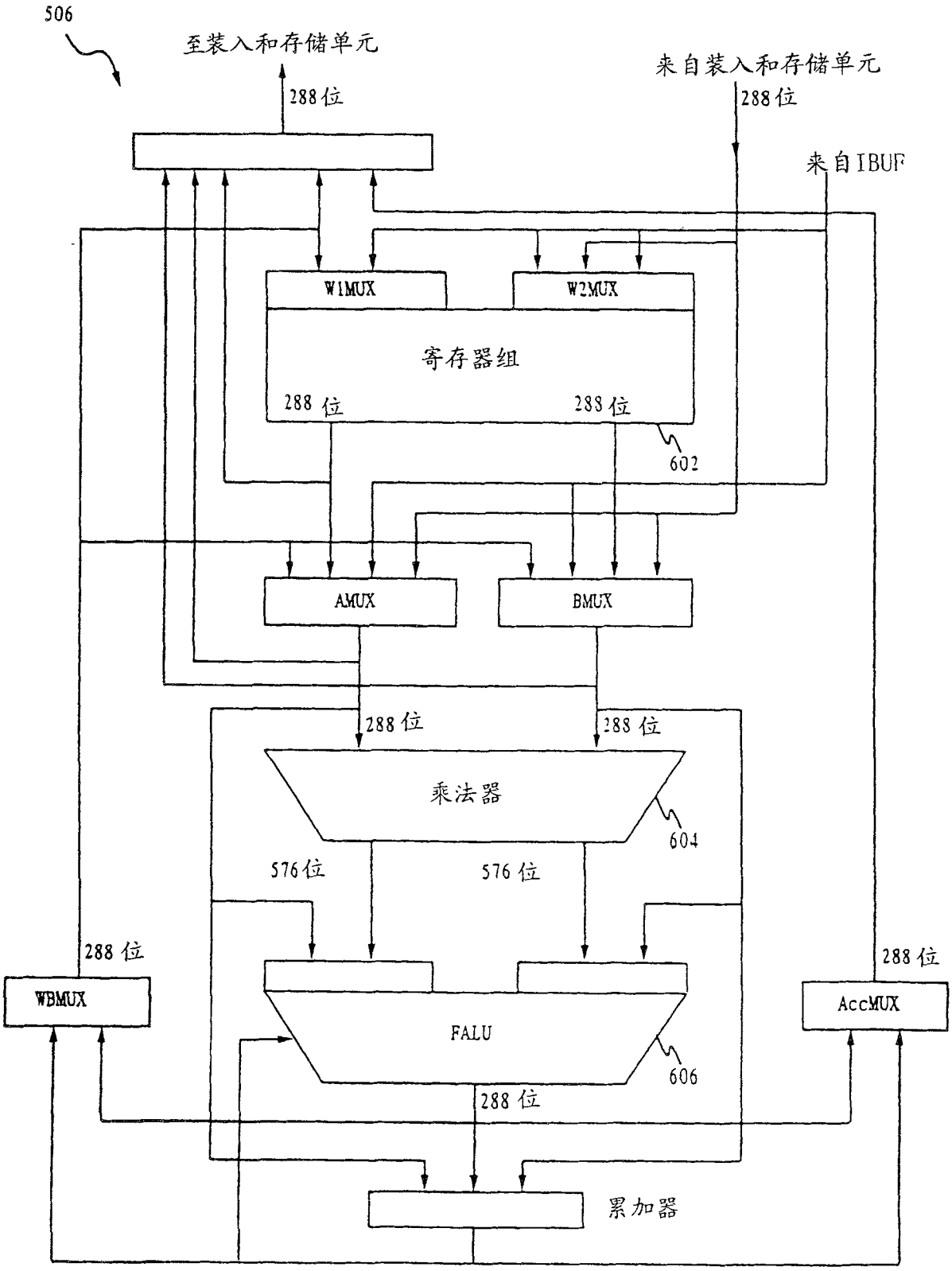

506
至装入和存储单元
288位
来自装入和存储单元
288位
来自IBUF
W1MUX
W2MUX
寄存器组
288 位
288 位
602
AMUX
BMUX
288 位
288 位
乘法器
604
576位
576 位
288 位
288 位
WBMUX
AccMUX
FALU
606
288 位
累加器

图 6

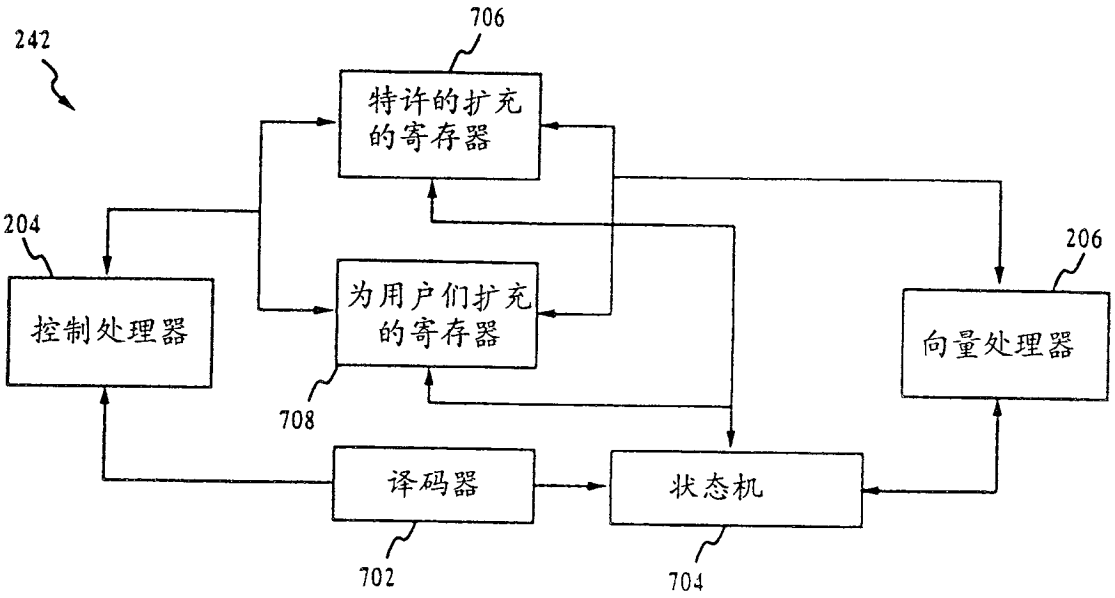

242
706
特许的扩充的寄存器
204
控制处理器
为用户们扩充的寄存器
708
206
向量处理器
译码器
状态机
702
704

图 7

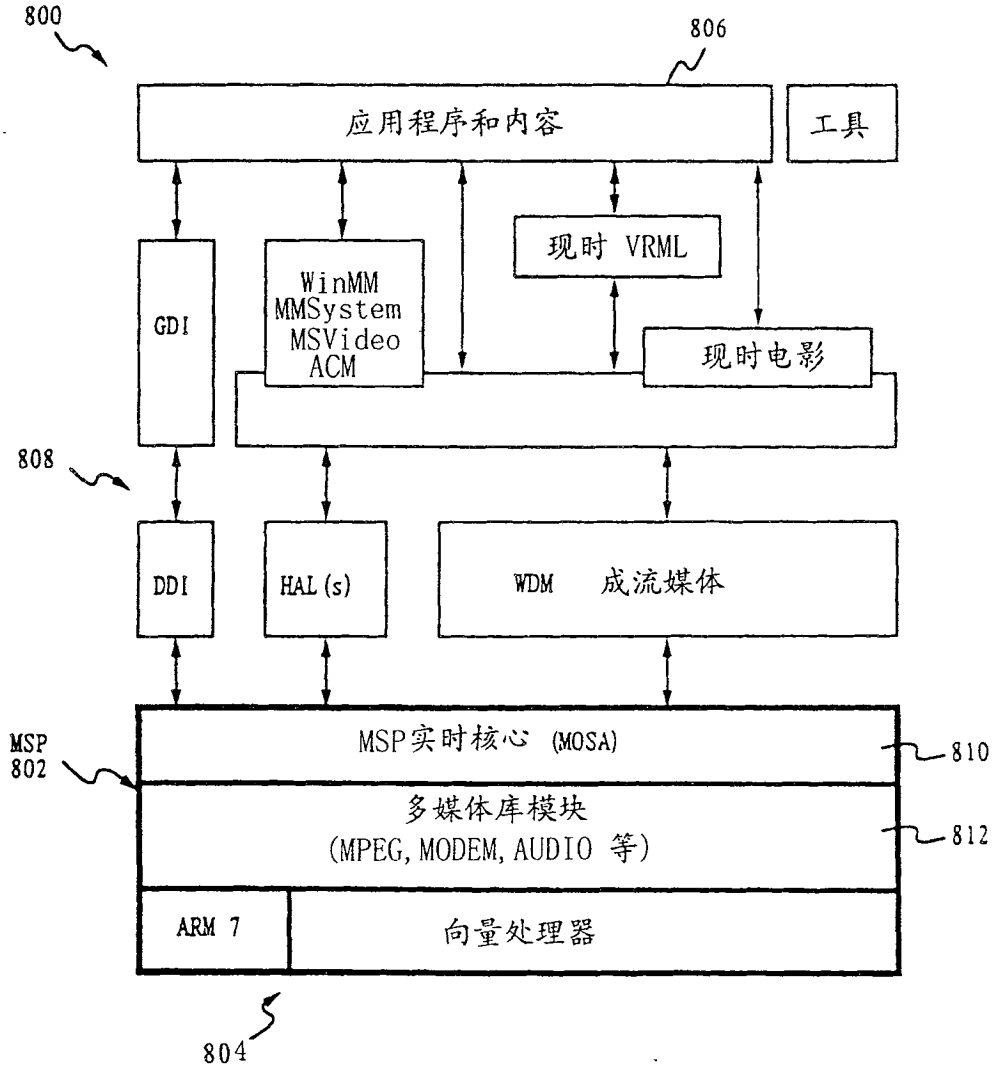

800
806
应用程序和内容
工具
现时 VRML
GDI
WinMM
MMSystem
MSVideo
ACM
现时电影
808
DDI
HAL(s)
WDM 成流媒体
MSP
802
MSP实时核心 (MOSA)
810
多媒体库模块
(MPEG, MODEM, AUDIO 等)
812
ARM 7
向量处理器
804

图 8

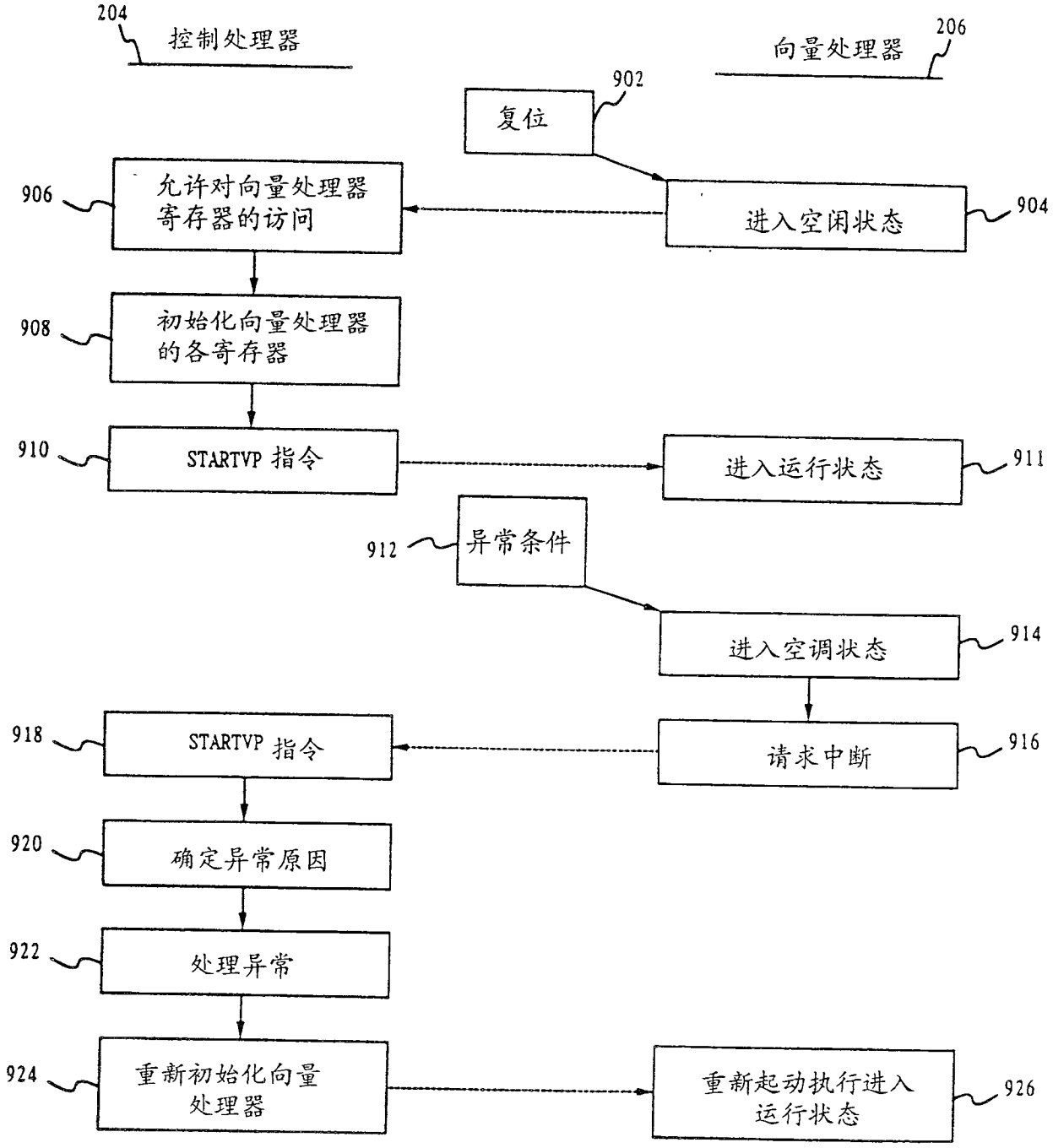

204
控制处理器
206
向量处理器
902
复位
906
允许对向量处理器寄存器的访问
904
进入空闲状态
908
初始化向量处理器的各寄存器
910
STARTVP 指令
911
进入运行状态
912
异常条件
914
进入空调状态
918
STARTVP 指令
916
请求中断
920
确定异常原因
922
处理异常
924
重新初始化向量处理器
926
重新起动执行进入运行状态

图 9